# Improving on Hidden Markov Models:
# An articulatorily constrained, maximum likelihood approach to speech recognition and speech coding

John Hogden
CIC-3, MS B265
Los Alamos National Laboratory
Los Alamos, NM  87545

## ABSTRACT

The goal of the proposed research is to test a statistical model of speech recognition that incorporates the knowledge that speech is produced by relatively slow motions of the tongue, lips, and other speech articulators. This model is called Maximum Likelihood Continuity Mapping (*Malcom*). Many speech researchers believe that by using constraints imposed by articulator motions, we can improve or replace the current hidden Markov model based speech recognition algorithms. Unfortunately, previous efforts to incorporate information about articulation into speech recognition algorithms have suffered because 1) slight inaccuracies in our knowledge or the formulation of our knowledge about articulation may decrease recognition performance, 2) small changes in the assumptions underlying models of speech production can lead to large changes in the speech derived from the models, and 3) collecting measurements of human articulator positions in sufficient quantity for training a speech recognition algorithm is still impractical. The most interesting (and in fact, unique) quality of *Malcom* is that, even though *Malcom* makes use of a mapping between acoustics and articulation, *Malcom* can be trained to recognize speech using only acoustic data. By learning the mapping between acoustics and articulation using only acoustic data, *Malcom* avoids the difficulties involved in collecting articulator position measurements and does not require an articulatory synthesizer model to estimate the mapping between vocal tract shapes and speech acoustics. Preliminary experiments that demonstrate that *Malcom* can learn the mapping between acoustics and articulation are discussed. Potential applications of *Malcom* aside from speech recognition are also discussed. Finally, specific deliverables resulting from the proposed research are described.

## I. Introduction

Hidden Markov models (HMM's) are among the most popular tools for performing computer speech recognition (see Huang, Ariki & Jack, 1990). One of the primary reasons that HMM's typically outperform other speech recognition techniques is that the parameters used for recognition are determined by the data, not by preconceived notions of what the parameters should be. This lets HMM's deal with intra- and inter-speaker variability despite our limited knowledge of how speech signals vary and despite our often limited ability to correctly formulate rules describing variability and invariance in speech. In fact, it is often the case that when HMM parameter values are constrained using our (possibly inaccurate) knowledge of speech, recognition performance decreases.

Nonetheless, many of the assumptions underlying HMM's are known to be inaccurate, and improving on these inaccurate assumptions within the HMM framework can be computationally expensive (Lee, 1989, p. 142). We argue that by using probabilistic models that more accurately embody the process of speech production, we can create models that have all the advantages of HMM's, but that should more accurately capture the statistical properties of real speech samples -- leading to more accurate speech recognition.

As reviewed elsewhere (McGowan & Faber, 1996; Rose, Schroeter & Sondhi, 1996) there have been several attempts to take advantage of articulation information to improve speech recognition. Some researchers have obtained improvements in speech recognition performance by building knowledge about articulation into HMM's (Deng & Sun, 1994; Erler & Deng, 1992), or by learning the mapping between acoustics and articulation using concurrent measurements of speech acoustics and human speech articulator positions (Papcun et al., 1992; Zlokarnik, 1995). Others have worked toward incorporating articulator information by using forward models (articulatory speech synthesizers) to study the relationship between speech acoustics and articulation (McGowan & Lee, 1996; Schroeter & Sondhi, 1994).

The model we will discuss, Maximum Likelihood Continuity Mapping (*Malcom*), differs from these previous attempts in that it learns the mapping from speech acoustics to articulator positions from acoustics alone -- articulator position measurements are not even used during training (the assertion that the mapping can be learned is backed up by the results of the preliminary experiments described below). Unlike Linear Predictive Coding (Markel & Gray, 1976; Wakita & Gray, 1975), which attempts to make the problem of recovering vocal tract shapes from speech acoustics tractable by using problematic simplifications (Sondhi, 1979, discusses several problems with the LPC approach), the assumptions underlying *Malcom* are well-founded. In fact, the main (and surprisingly powerful) assumption used by *Malcom* is that articulator motions produced by muscle contractions have little energy above 15 Hz, which is well documented (Muller & McLeod, 1982; Nelson, 1977).

The fact that *Malcom* derives so much about the relationship between acoustics and articulation from so few assumptions should be a big advantage over current systems. Consider that as we build more assumptions about articulator motions into existing models, we have a greater chance of incorporating invalid constraints and potentially decreasing recognition performance. For example, some of the rules used in existing systems "are rather simplistic and contain several unrealistics aspects" (Deng & Sun, 1994, p. 2717). Furthermore, systems that require that information about articulation be learned from human data suffer from the fact that collecting concurrent measurements of speech acoustics and articulator positions in enough quantity to train a large vocabulary speech recognition algorithm is currently impractical. Even systems that avoid the need for human data by using articulatory speech synthesizers may be adversely affected by inaccurate assumptions -- the mapping between speech acoustics and speech articulation for articulatory speech synthesizers is strongly dependent on assumptions underlying the synthesizers and appears to differ in important ways from the mapping observed for human speech production (Hogden et al., 1996).

Before discussing how to use articulatory constraints, we will give a brief description of HMM's. This will allow us to highlight the similarities and differences between HMM's and the proposed technique. In a straightforward implementation of the HMM approach, models are made of each word in the vocabulary. The word models are constructed such that we can determine the probability that any acoustic speech sample would be produced given a particular word model. The word model most likely to have created a speech sample is taken to be the model of the word that was actually spoken. For example, suppose we produce some new speech sample, **Y**. If $w_i$ is the model for word *i*, and $w_i$ maximizes the probability of **Y** given $w_i$, then a HMM speech recognition algorithm would take word *i* to be the word that was spoken. In other variants of HMM speech recognition, models are made of phonemes, syllables, or other subword units.

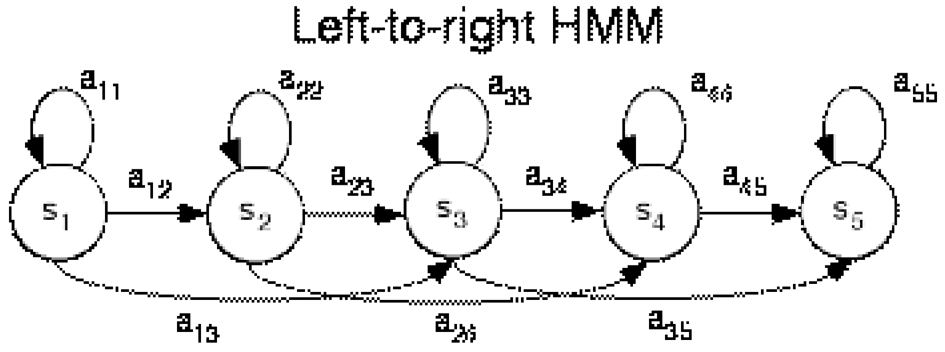

Figure 1

Figure 1 shows a 5 state HMM of a type commonly used for speech recognition (Rabiner & Juang, 1986). Each of the circles in Figure 1 represents a HMM state. At any time, the HMM has one active state and a sound is assumed to be emitted when the state becomes active. The probability of sound $y$ being emitted by state $s_i$ is determined by some parameterized distribution associated with state $s_i$ (e.g. a multivariate Gaussian parameterized by a mean and a covariance matrix). The connections between the states represent the possible interstate transitions. For example, in the left-to-right model below, if the model is in state $s_2$ at time $t$, then the probability of being in state $s_4$ at time $t+1$ is $a_{24}$.

HMM's are trained using a labeled speech data base. For example, the data set may contain several samples of speakers producing the word "president". Using this data, the parameters of the "president" word model (the transition probabilities and the state output probabilities) are adjusted to maximize the likelihood that the "president" word model will output the known speech samples. Similarly, the parameters of the other word models are also adjusted to maximize the likelihood of the appropriate speech samples given the models. We expect that as the word models more closely match the distributions of actual speech samples (i.e. the probability of the data given the word models increases), the recognition performance will improve -- which is why the models are trained in the first place.

*Malcom* attempts to make the word models give better estimates of the distributions of speech data by basing the models on the actual processes underlying speech production. Consider that speech sounds are produced by slowly moving articulators. Thus, if we know the relationship between articulator positions and speech acoustics, we should be able to use information about the articulator positions preceding time $t$ to accurately predict the articulator positions at time $t$, and therefore predict the acoustic signal at time $t$. In the following discussion, we show how maximum likelihood techniques can be brought to bear on this problem.

## II. *Malcom*

As with HMM's, in order to determine which sequence of words was most likely to have created the observed data, we want to be able to determine the probability of the observed data given a word model. In the articulatory recognition algorithm presented here, each word is described in terms of the sequence of articulator positions used to create the word. Thus, to find the model for the word "president", we collect several acoustic recordings of "president" and then find the articulator path that maximizes the conditional probability of the recorded acoustics sequences. In section II.A we describe how an articulator path (a word model) that maximizes the probability of the data can be found if we know about the mapping from speech sounds to articulator positions.

Because we cannot determine the optimal articulator trajectories without knowing the mapping from articulator positions to acoustics, in section II.B we show how *Malcom* learns an invertable, probabilistic mapping between articulator positions and acoustics. Acoustic speech signals are represented as sequences of vector quantization (VQ) codes (Ahalt, Krishnamurthy, Chen & Melton, 1990; Gray, 1984; Linde, Buzo & Gray, 1980 describe vector quantization). In section II.C we show an example of how *Malcom* can be applied to the case where the distribution of articulator positions that produce a code is assumed to be Gaussian.

### II.A Finding Articulatory Trajectories that Maximize the Probability of the Observed Data

In order to describe how to use *Malcom* to perform speech recognition, we will start with some definitions. Let:

*n* = the number of vector quantization codes in a given speech sample,

*c(t)* = the VQ code assigned to the $t^{\text{th}}$ window of speech,

**c** = [*c(1)*, *c(2)*, ... *c(n)*] = a sequence of VQ codes used to describe a speech sample,

$x_i(t)$ = the position of articulator *i* at time *t,*

$\mathbf{x}(t) = [x_1(t), x_2(t), \ldots x_d(t)]$ = a vector composed of the positions of all the articulators at time *t*, and

$\mathbf{X} = [\mathbf{x}(1), \mathbf{x}(2), \ldots \mathbf{x}(n)]$ = a sequence of articulator configurations.

Further definitions are needed to specify the mapping from articulation to VQ codes. Let

$P(c_i)$ = the probability of observing code $c_i$ given no information about context,

$P(\mathbf{x}|c_i,\square)$ = the probability that articulator position **x** was used to produce VQ code $c_i$ where:

□ = a set of model parameters, e.g. □ could include the mean and covariance matrix of a Gaussian probability density function used to model the distribution of **x** given *c*.

Note that we have not specified distributions that give $P(\mathbf{x}|c_i,\square)$. We avoided limiting $P(\mathbf{x}|c_i,\square)$ because we want to allow for the various possible mappings from acoustics to articulator positions. For example, it has often been argued that many different articulator positions can be used to produce the same acoustic signal (Atal, Chang, Mathews & Tukey, 1978; Schroeter & Sondhi, 1994). Although the limited research on human speech production data argues that articulator positions can be recovered from acoustics much more accurately than computer simulations suggest (Hogden et al., 1996; Ladefoged, Harshman, Goldstein & Rice, 1978; Papcun et al., 1992), if there are multimodal distributions of articulator positions that can be used to produce identical acoustic signals, then it may be necessary to specify $P(\mathbf{x}|c_i,\square)$ as a mixture of Gaussians.

With these definitions, the probability of observing code $c_j$ given that the current articulator position is **x**, is expressed as:

$$P(c_j|\mathbf{x},\square) = \frac{P(c_j,\mathbf{x}|\square)}{P(\mathbf{x}|\square)} = \frac{P(c_j,\mathbf{x}|\square)}{\underset{i}{\square}\, P(c_i,\mathbf{x}|\square)} = \frac{P(\mathbf{x}|c_j,\square)P(c_j)}{\underset{i}{\square}\, P(\mathbf{x}|c_i,\square)P(c_i)} \qquad \text{Eq. 1}$$

Assuming conditional independence

$$P\left[\mathbf{c}|\mathbf{X},\square\right] = \overset{n}{\underset{t=0}{\square}}\, P\left[c(t)|\mathbf{x}(t),\square\right] \qquad \text{Eq. 2}$$

Note that the probability of observing a code is not assumed to be independent of the preceding and subsequent codes, it is only assumed to be conditionally independent. So if **x***(t)* is dependent on **x***(t')* then *c(t)* is dependent on *c(t')*. As demonstrated below, by using an appropriately constrained model of possible articulator trajectories the sequences of codes can be tightly constrained in a biologically plausible manner.

It is possible to find the articulator path that maximizes the probability of a sequence of codes, i.e. find the **X** that maximizes $P(\mathbf{c}|\mathbf{X},\square)$, or equivalently, that maximizes $LogP\left[\mathbf{c} \mid \mathbf{X},\square\right]$. Taking the logarithm of Eq. 2 we get:

$$LogP\left[\mathbf{c}\middle|\mathbf{X},\square\right] = \sum_t LogP\left[c(t)\middle|\mathbf{x}(t),\square\right] \qquad \text{Eq. 3}$$

To show how to maximize $P(\mathbf{c}|\mathbf{X},\square)$, we substitute Eq. 1 into Eq. 3 and separate the terms in the logarithm to get:

$$LogP\left[\mathbf{c}\middle|\mathbf{X},\square\right] = \sum_t \left\{ LogP\left[\mathbf{x}(t)\middle|c(t),\square\right] + LogP\left[c(t)\right] - Log\sum_i P\left[\mathbf{x}(t)\middle|c_i,\square\right]P\left[c_i\right]\right\} \qquad \text{Eq. 4}$$

Using $\nabla$ to denote the gradient with respect to the components of $\mathbf{x}(t')$, $LogP\left[\mathbf{c}\middle|\mathbf{X},\square\right]$ is maximized when:

$$\nabla LogP\left[\mathbf{c}\middle|\mathbf{X},\square\right] = 0 \quad \forall t' \qquad \text{Eq. 5}$$

Substituting Eq. 4 for the left hand side of Eq.5 and reducing gives:

$$\nabla \sum_t \left\{ LogP\left[\mathbf{x}(t)\middle|c(t),\square\right] + LogP\left[c(t)\right] - Log\sum_i P\left[\mathbf{x}(t)\middle|c_i\right]P\left[c_i\right]\right\} = 0 \quad \forall t'$$

$$\sum_t \left\{ \frac{\nabla P\left[\mathbf{x}(t)\middle|c(t),\square\right]}{P\left[\mathbf{x}(t)\middle|c(t),\square\right]} + \frac{\nabla P\left[c(t)\right]}{P\left[c(t)\right]} - \frac{\sum_i \nabla P\left[\mathbf{x}(t)\middle|c_i,\square\right]P\left[c_i\right]}{\sum_i P\left[\mathbf{x}(t)\middle|c_i,\square\right]P\left[c_i\right]} \right\} = 0 \quad \forall t'$$

concluding with:

$$\nabla LogP\left[\mathbf{c}\middle|\mathbf{X},\square\right] = \frac{\nabla P\left[\mathbf{x}(t')\middle|c(t'),\square\right]}{P\left[\mathbf{x}(t')\middle|c(t'),\square\right]} - \frac{\sum_i P\left[c_i\right]\nabla P\left[\mathbf{x}(t')\middle|c_i,\square\right]}{\sum_i P\left[\mathbf{x}(t')\middle|c_i,\square\right]P\left[c_i\right]} = 0 \quad \forall t' \qquad \text{Eq. 6}$$

The preceding analysis is incomplete because it ignores constraints on the possible articulator paths. To incorporate biologically plausible constraints on articulator motion, we will allow only those articulator trajectories that have all their energy below some cut-off frequency (say 15 Hz, since actual articulator paths have very little energy above 15 Hz). The constraint that the articulator path have all of its energy below the cut-off frequency is equivalent to requiring that the path lie on a hyperplane composed of the axes defined by low frequency sine and cosine waves.

When $\nabla$Log($\mathbf{c}|\mathbf{X},\square$) is perpendicular to the constraining hyperplane, so that Log($\mathbf{c}|\mathbf{X},\square$) can not increase without **X** traveling off the hyperplane, then we have reached a constrained local minimum (Marsden & Tromba, 1981). Thus, the smooth path that maximizes the likelihood of the observed data is the path for which $\nabla$Log($\mathbf{c}|\mathbf{X},\square$) has no components with energy

below the cut-off frequency. This suggests the following algorithm for finding the smooth path that maximizes the probability of the data:
1) start with any smooth articulator path.
2) find the gradient of the log probability of the smooth path.
3) low-pass filter the gradient to get the gradient projected onto the constraining hyperplane.
4) add the low-pass filtered gradient times some small constant to the path to get a better estimate of the most likely smooth path.
5) repeat steps 2 - 4 until the algorithm converges.

There are also a variety of standard numerical algorithms, such as the conjugate gradient technique, that can be used to maximize functions. Using one of these algorithms can speed up the process of finding the most likely smooth path.

II.B Finding a Mapping from Articulation to Acoustics

In the preceding section, we assumed that we knew $P(c)$ and $P(\mathbf{x}|c,\square)$. In this section we show that these values can be determined using only acoustic data. This is the most important section of this proposal, because $P(\mathbf{x}|c,\square)$ is a probabilistic mapping from speech sounds to articulator positions, and our claim is that this mapping can be inferred using training data composed of only sequences of VQ codes. The techniques in this section allow us to use articulator information without hard-wiring possibly faulty knowledge of phonetics into a model, without collecting measurements of articulator positions, and without using potentially inaccurate articulatory synthesizers to learn the mapping from acoustics to articulator positions. We develop the techniques in this section and demonstrate their power in the preliminary experiment below.

Using maximum likelihood estimation, it is possible to find a good approximation of the relationship between acoustics and articulation by building on the framework presented above. All we have to do is iteratively repeat two steps:

1) given a collection of quantized speech signals and some initial estimate of the mapping from acoustics to speech, use the procedures in section II.A to find the paths that maximize the conditional probability of the observed data.

2) given the paths that maximize the probability of the data, find the value of □ and the $P(c_i)$ values that will maximize (or at least increase) the conditional probability of the data.

Since both of these steps will increase the probability of the data, by iteratively repeating them, we will increase the probability of the data until we have reached a local (possibly global) maximum.

Calculation of $P(c)$ and □ can be accomplished using standard maximization algorithms. Maximization algorithms that use gradient information typically require less computation than algorithms that don't use the gradient, making it advantageous to have expressions for □Log($\mathbf{c}|\mathbf{X}$,□) with respect to □ and with respect to $P(c)$. The expression for □Log($\mathbf{c}|\mathbf{X}$,□) with respect to □ is:

$$\Box LogP\left[\mathbf{c}|\mathbf{X},\Box\right]$$

$$= \Box \sum_t \Box LogP\left[\mathbf{x}(t)|c(t),\Box\right] + LogP\left[c(t)\right] \Box Log \sum_i P\left[\mathbf{x}(t)|c_i\right]P\left[c_i\right] \Box$$

$$= \sum_t \Box \frac{\Box P\left[\mathbf{x}(t)|c(t),\Box\right]}{P\left[\mathbf{x}(t)|c(t),\Box\right]} + \frac{\Box P\left[c(t)\right]}{P\left[c(t)\right]} \Box \frac{\sum_i \Box\left\{P\left[\mathbf{x}(t)|c_i,\Box\right]P\left[c_i\right]\right\}}{\sum_i P\left[\mathbf{x}(t)|c_i,\Box\right]P\left[c_i\right]} \Box$$

concluding with:

$$\Box LogP\left[\mathbf{c}|\mathbf{X},\Box\right] = \sum_t \Box \frac{\Box P\left[\mathbf{x}(t)|c(t),\Box\right]}{P\left[\mathbf{x}(t)|c(t),\Box\right]} \Box \frac{\sum_i \Box\left\{P\left[\mathbf{x}(t)|c_i,\Box\right]P\left[c_i\right]\right\}}{\sum_i P\left[\mathbf{x}(t)|c_i,\Box\right]P\left[c_i\right]} \Box \qquad \text{Eq. 7}$$

The expression for □Log(**c**|**X**,□) with respect to *P(c)* can be derived as follows:

$$\Box LogP\left[\mathbf{c}|\mathbf{X},\Box\right] = \sum_{t \Box c(t)=c_k} \frac{1}{P\left[c_k\right]} \Box \sum_{t=0}^{n} \Box \frac{P\left[\mathbf{x}(t)|c_k,\Box\right]}{\sum_i P\left[\mathbf{x}(t)|c_i,\Box\right]P\left[c_i\right]} \Box$$

$$= \frac{n_k}{P\left[c_k\right]} \Box \sum_{t=0}^{n} \Box \frac{P\left[\mathbf{x}(t)|c_k,\Box\right]}{\sum_i P\left[\mathbf{x}(t)|c_i,\Box\right]P\left[c_i\right]} \Box$$

$$= \frac{n_k}{P\left[c_k\right]} \Box \frac{1}{P(c_k)} \sum_t P\left[c_k|\mathbf{x}(t),\Box\right] \qquad \text{Eq. 8}$$

$$\Box \frac{n_k}{P\left[c_k\right]} \Box \frac{1}{P(c_k)} \sum_{\mathbf{x}} P(\mathbf{x})P\left[c_k|\mathbf{x},\Box\right]$$

$$= \frac{n_k}{P\left[c_k\right]} \Box 1$$

where $n_k$ is the number of times $c_k$ is observed in the speech sample. Since the sum of the *P(c)* values must be 1, finding the *P(c)* values that maximize the conditional probability of the data is a constrained optimization problem in which we find the *P(c)* values by solving:

$$\frac{n_k}{P(c_k)} \Box 1 = \Box\Box \sum_i P(c_i) = \Box \qquad \text{Eq. 9}$$

From Eq. 9 we see that setting

$P(c_k) = n_k/n$ Eq. 10

will maximize the conditional probability of the data.

Thus, using only acoustic speech samples, two important learning steps can be performed. The first is to learn the relationship between acoustics and articulation. The second is to learn articulator trajectory models for each word. However, the data required for the first learning step is not necessarily the same as that required for the second learning step. We do not need labeled training data to learn the relationship between acoustics and articulation for a speaker but we do need labelled training data to learn word models.

## II.C Example

Although many different forms of the $P[\mathbf{x}(t)|c(t),\square]$ distributions could be used, in this section we will derive the gradient equations for the case where the distribution of articulator positions that produce sounds quantized by code *c* is a multivariate Gaussian characterized by the equation:

$$P[\mathbf{x} \mid c,\square] = \frac{1}{(2\square)^{d/2}|\square(c)|^{1/2}}\exp\square\square\frac{1}{2}[\mathbf{x}\square\square(c)]^t\square^{\square 1}(c)[\mathbf{x}\square\square(c)]\square \qquad \text{Eq. 11}$$

where:

*d* is the number of dimensions in the articulator space (i.e. the number of articulators),

$\square(c)$ is a vector giving the mean of all the articulator positions used to produce sounds quantized with vector quantization code *c*. For example, $\square_i(c)$ may give the mean lower lip position used to create sounds quantized as code *c*,

$\square(c)$ is the covariance matrix of the multivariate Gaussian distribution of articulator positions that produce sounds quantized with code *c*, and

$\mathbf{x}$ is a vector describing an articulator configuration.

Before we derive the gradient equations, note that the **x**, □(c) and □(c) values that maximize the conditional probability of the data are not unique. For example suppose we have found the **x**, □(c), and □(c) values that maximize the conditional probability of the data. Let **R** be an orthogonal rotation matrix, **y** be an arbitrary vector, and □ be a nonzero scalor. If we let **x**□=□**Rx**+**y**, let □□(c)=□**R**□(c)+**y**, and let □□(c) = $\square^2$**R**□(c)$\mathbf{R}^t$ then the probability of **x**□given a code and the model is

$$P[\mathbf{x}\square| c,\square\square] = \frac{1}{\square^{d/2}}P[\mathbf{x} \mid c,\square] \qquad \text{Eq. 12}$$

If we substitute Eq. 12 into Eq. 1 we find that the conditional probability of the VQ codes is the same for **x**□, □□(c) and □□(c) as it was for **x**, □(c) and □(c). Thus we can rotate, reflect, translate and scale the □(c) values and get an equally good solution as long as we make the appropriate changes to the **x** and □(c) values.

Returning to the problem of finding the gradient equations for the Gaussian probability density function, let □ denote the gradient with respect to the components of **x**, so

$$\square P[\mathbf{x}|c,\square] = \square P[\mathbf{x}|c,\square]\square^{\square 1}(c)[\mathbf{x}\square\square(c)] \qquad \text{Eq. 13}$$

Which can be substituted into Eq. 6 to show how to find the path that maximizes the conditional probability of the data:

$$\nabla LogP\left[\mathbf{c}\middle|\mathbf{X},\phi\right] = -\Sigma^{-1}\left[c(t)\right]\left\{\mathbf{x}(t) - \mu\left[c(t)\right]\right\}$$

$$+\frac{\sum_{i} P\left[c_i\right] P\left[\mathbf{x}(t)\middle|c_i,\phi\right]\Sigma^{-1}(c_i)\left\{\mathbf{x}(t) - \mu(c_i)\right\}}{\sum_{i} P\left[\mathbf{x}(t)\middle|c_i,\phi\right]P\left[c_i\right]}$$

Eq. 14

Similarly, the gradient with respect to $\mu(c_k)$ is:

$$\nabla P\left[\mathbf{x}\middle|c_i,\phi\right] = P\left[\mathbf{x}\middle|c_i,\phi\right]\Sigma^{-1}(c_i)\left[\mathbf{x} - \mu(c_i)\right]\delta_{ik} \qquad \delta_{ik} = \begin{cases}1 & if\ i = k\\ 0 & if\ i \neq k\end{cases}$$

Eq. 15

Which, finally, can be substituted into Eq. 7 to get:

$$\nabla LogP\left[\mathbf{c}\middle|\mathbf{X},\phi\right] = \sum_{t \ni c(t)=c_k} \Sigma^{-1}\left[c(t)\right]\left\{\mathbf{x}(t) - \mu\left[c(t)\right]\right\}$$

$$-\sum_{t}\frac{P\left[c_k\right]P\left[\mathbf{x}(t)\middle|c_k,\phi\right]\Sigma^{-1}(c_k)\left\{\mathbf{x}(t) - \mu(c_k)\right\}}{\sum_{i} P\left[\mathbf{x}(t)\middle|c_i,\phi\right]P\left[c_i\right]}$$

Eq. 16

## III. Preliminary Studies

### III.A Recovering Mean Articulator Positions From Acoustics

Preliminary studies have demonstrated that a simplification of *Malcom* (*Malcom 1*) is able to learn the mapping from acoustics to articulator position using a training set composed only of acoustic data. Specifically, this experiment shows that, if we categorize short windows of speech acoustics, and make the assumption that the articulator positions used to create a given acoustic category are distributed according to a multivariate Gaussian, we can estimate the mean articulator configuration associated with each acoustic category. This experiment is a shortened version of an experiment reported elsewhere (Hogden, 1995b).

The simplifications in *Malcom 1* significantly decrease training time when trying to learn the mapping from acoustics to articulation, but do not allow the direct use of *Malcom 1* for estimating the conditional probability of the data. Nonetheless, learning the mapping from acoustics to articulation is considered an extremely difficult task, so the fact that *Malcom 1* succeded in learning this mapping without training on articulator measurements can be taken as an indication of the power of *Malcom.*

### III.B Data

All speech samples were produced by a male Swedish speech scientist fluent in both Swedish and English. The speaker produced utterances containing two vowels spoken in a /g/ context with a continuous transition between the vowels, as in /guog/. The vowels in the utterances were all pairs of 9 Swedish vowels (/i/, /e/, /æ/, /a/, /o/, /u/, and the front rounded vowels /y/, /ɯ/, and /ø/), as well as the English vowel /ɛ/, for a total of 90 utterances (Fant, 1973). While recording the utterances, the positions of receiver coils on the tongue, jaw, and lips were measured using an EMMA system (Perkell et al., 1992). Note that the articulator positions were only measured in order to allow comparisons between estimated and actual articulator positions.

## III.C Signal Processing

Spectra were recovered from 32 cepstrum coefficients of 25 ms Hamming windows of speech These spectra were categorized into 256 categories using vector quantization and the mean articulator configuration associated with each code was calculated.

## III.D Calculating Actual Mean Articulator Positions

While *Malcom 1* estimates the mean articulator configurations without articulatory measurements, in order to compare *Malcom 1's* estimates with the actual mean articulator configurations, it is necessary to calculate the mean articulator configurations from the articulator measurements. The mean articulator position associated with sound type 1 was found by averaging the receiver coil configurations used to produce sounds that were classified as type one. The mean articulator position was calculated for each other sound type in the same way.

## III.E Estimating the Mean Articulator Positions Using *Malcom 1*

Instead of maximizing the conditional probability of the observed data, *Malcom 1* recover the mapping between acoustics and articulation by maximizing the probability of the smooth articulator paths. Mathematically, this amounts to ignoring the second term in Eqs. 14 & 16. The simplified versions of Eqs. 14 and 16 are, respectively:

$$\square LogP\left[\mathbf{c}\middle|\mathbf{X},\square\right] = \square\,\square^{\square 1}\left[c(t\square)\right]\left\{\mathbf{x}(t\square)\,\square\,\square\left[c(t\square)\right]\right\} \qquad \text{Eq. 17}$$

and

$$\square LogP\left[\mathbf{c}\middle|\mathbf{X},\square\right] = \underset{t}{\square}\;\square^{\square 1}\left[c(t)\right]\left\{\mathbf{x}(t)\,\square\,\square\left[c(t)\right]\right\} \qquad \text{Eq. 18}$$

Notice that Eq. 18 is significantly simpler to maximize than Eq. 16. Eq. 16 requires an iterative maximization algorithm whereas Eq. 18 can be solved analytically. The analytic solution for Eq. 18 is to set

$$\square(c_i) = {\sum\limits_{t\,\square\, c(t)=c_i}}^{\!\!\square}\mathbf{x}(t) \Big/ n_i \qquad \text{Eq. 19}$$

*P(c)* is not calculated in *Malcom 1* because no information about *P(c)* can be extracted without trying to maximize the conditional probability of the data instead of the probability of the smooth paths. For this study, all the covariance matrices were set to the identity matrix.

One difficulty in using *Malcom 1* is that there is a degenerate solution in which all of the $\square(c)$'s converge to the same point. This problem was avoided by forcing the variance of the $\square(c)$'s to be 1.

## III.F Comparing Estimated to Actual Mean Articulator Configurations

Since the mean values calculated by *Malcom 1* can be translated, rotated, reflected or scaled compared to the actual mean articulator positions, comparisons between the estimated mean articulator positions and the actual mean articulator positions is non-trivial. One way to determine whether the estimates of the mean articulator positions in a maximum likelihood continuity map supply information about the actual mean articulator positions is to see whether equations can be constructed giving the actual mean positions from the estimated mean positions. In order for the mean articulator position estimates to be useful, the equations should be simple. This experiment focused on linear functions of the form:

$$\hat{A}_{ic} = \sum_{d=1}^{D} \square_{id} m_{dc} + k_i \quad \text{with } \square_{ic} = A_{ic} \square \hat{A}_{ic} \qquad \text{Eq. 20}$$

where:

$\hat{A}_{ic}$ is the mean position of the receiver coil *i* for sounds of type *c* as estimated by the linear equation,

$A_{ic}$ is the actual mean position of the receiver coil *i* for sounds of type *c*,

$D$ is the number of dimensions in the *Malcom 1* solution,

$m_{dc}$ is the position of code *c* on the $d^{th}$ dimension of the *Malcom 1* solution.

The other parameters, $\square_{id}$ and $k_i$, are values that will minimize the sum of the squared error terms. An equation of this form is particularly interesting because solving for the unknown $\square_{id}$ and $k_i$ values is equivalent to finding axes in the *Malcom 1* solution that correspond most closely to the articulator positions -- essentially compensating for the fact that the *Malcom 1* solution can be rotated, scaled, translated, or reflected with respect to the actual articulator positions.

Using standard multiple regression techniques (Neter, Wasserman & Kutner, 1985) the $\square_{id}$ and $k_i$ values that minimize the sum of the squared error terms can be found. Multiple regression also gives a quantitative measure of the extent to which the equation is accurate, namely, the multiple regression r value.

Figure 2 shows the multiple regression r values obtained when trying to relate the positions of codes in the maximum likelihood continuity map to the mean articulator positions of three key articulators -- the tongue rear (*x* and *y* positions), the tongue tip (*y* position) and the upper lip (*y* position). Figure 2 shows that a four dimensional *Malcom 1* solution is sufficient to capture much of the information about the mean articulator positions, and that *Malcom 1* solutions with more than four dimensions do only slightly better than a four dimensional solution. Figure 2 also shows that tongue body positions can be recovered surprisingly accurately (Pearson r values of around 95%).

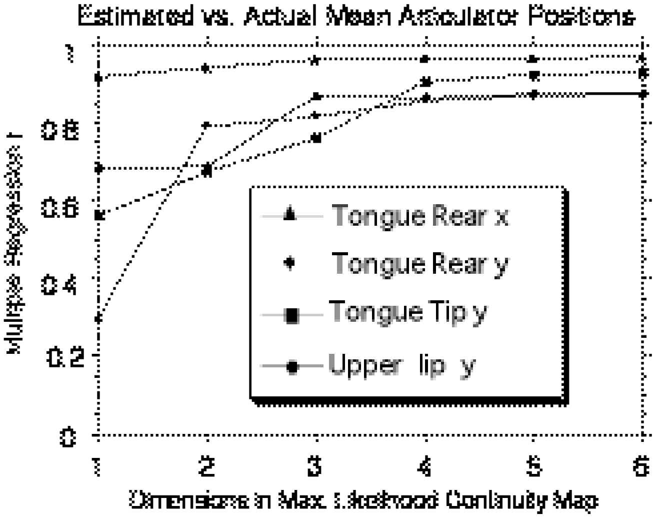

Figure 2

III.G Using Mean Articulator Configurations to Estimate Actual Articulator Configurations
The mean of all the articulator configurations used to produce an acoustic segment is not necessarily a good estimate of the actual articulator used to produce a segment. For example, if two very different articulator positions (call the positions 1 and 2) create the same acoustic signal (call it signal type 3), but articulator configurations between positions 1 and 2 produce different signals, then the average articulator configuration will not even be among those that create the signal type 3. (Jordan & Rumelhart, 1992). However, since we have both acoustic and articulator measurements in the data set, it is possible to determine whether the mean articulator positions are good estimates of the actual articulator positions. In short, the mean articulator positions are good estimates of the actual articulator positions for this data set -- root mean squared error values for points on the tongue were less than 2 mm (Hogden et al., 1996). Of course, articulation positions can be recovered more accurately from acoustics when a small articulator motion creates a large change in acoustics, e.g. near constrictions.

## IV. Potential Applications of *Malcom* (other than speech recognition)

Speech recognition is already a 500 million to 1 billion dollar/year industry, despite limitations of the current tools (Mott, 1995). A sufficiently good speech recogniton algorithm could completely change the way people interact with computers, possibly doubling the input rate, since the number of words spoken per minute is more than double the average typing rate. So improving speech recognition is an admirable goal in itself. But, the impact of *Malcom* could extend far beyond speech recognition. *Malcom* is a relatively general statistical technique that has a variety of potential speech and non-speech applications. To give an impression of the possible payoff of developing *Malcom*, we briefly describe some of these applications below.

IV.A Speaker Recognition
Presumably, if *Malcom* leads to improvements in speech recognition it should also improve speaker verification/identification algorithms, since techniques used for speaker verification are very similar to those used for speech recognition. To use *Malcom* for speaker recognition, different mappings from acoustics to articulation would be made for each speaker. The likelihood that any given speaker produced a new speech sample could be calculated using the technique described in section II.A. For speaker identification, the speaker most likely to have produced the speech signal would be chosen. For speaker

verification, the speaker would be verified if the likelihood of producing the speech was sufficiently high or if it was higher than some cohort set.

High performance speech recognition would not only have a wide variety of commercial uses (e.g. preventing unauthorized telephone access to bank accounts) but could be important for controlling access to classified information. The advantage of using voice characteristics to verify identity is that voice characteristics are the only biometric data that are typically transmitted over phone lines.

IV.B Speech Coding

The path which maximizes the conditional probability of the data is also the path that minimizes the number of bits that need to be transmitted in addition to the smooth path to specify the data. This can be seen from information theory (Sayood, 1996), which shows that the number of bits that must be transmitted in addition to the smooth path is:

$$\text{bits} = \sum_t \left(1 - Log P\left[c(t) \middle| \mathbf{x}(t), \square\right]\right) = \sum_t 1 - \sum_t Log P\left[c(t) \middle| \mathbf{x}(t), \square\right] \qquad \text{Eq. 21}$$

Since we are maximizing $\sum_t Log P\left[c(t) \middle| \mathbf{x}(t), \square\right]$, we are minimizing the number of bits.

Furthermore, articulatory trajectories can be transmitted with many fewer bits/second than speech sounds. Consider that the position of a single articulator can be transmitted using about 30 samples/second and the range of articulator positions is much smaller than the range of amplitudes found in acoustic signals. So, assume that we need about 5 bits per sample (similar to what is needed for LPC coefficients) for the tongue body x and y coordinates, the tongue tip, and for two lip parameters, but only 1 bit per sample for the velum (it is either opened or closed). Further assume that we need to transmit about 600 bits/second for pitch, voicing, and gain information (as in the 2.4 kbit/second U.S. Government Standard LPC-10). This gives us an estimate of about 1380 bits/second, or about 40% less than the 2.4 kbit/second U.S. Government Standard LPC-10. These considerations suggest that this approach has potential as a low bit-rate speech transmission technique (c.f. Schroeter & Sondhi, 1992).

Such an application is likely to be particularly valuable in satellite communication. To judge the value, consider that the INMARSAT (A) provides communications service at a rate of $39.00 per kbps-hour, based on figures provided in a January 1996 AFCEA (102) on Military Satellite Communications. Arbitrarily, for only six hours of voice communication per day for a year at 2400 bps, the cost of this service is $204,984. Let me emphasize that this estimate is for only one voice channel. Even a relatively moderate (say 20%) decrease in the number of bits per second needed to transmit speech would be worth approximately $40,000 per voice per year. Furthermore, judging from recent experiments with speech synthesis (see section IV.C below) speech reconstructed using this technique is likely to be higher quality than the government standard.

IV.C Speech Synthesis

Recent results show that HMM's can be used to produce high quality synthesized speech (Donovan, 1996). However, since the HMM model of speech transitions is unrealistic, *Malcom* can likely be used in much the same way as HMM's to produce higher quality synthesized speech.

In addition, since it should be easier to describe words in terms of the articulator motions that produce the words than by describing the sound waves that are produced, *Malcom* may simplify the user interface for speech synthesizers. For example, we may be able to draw an articulator path and use a *Malcom* derived mapping from articulator positions to acoustics to produce synthesized speech.

IV.D Voice Mimicry

It may be possible to have a person speak into a computer, convert the speech sounds to articulator trajectories, and then synthesize a different person's voice with trajectories from the first person (Hogden, 1995a) -- essentially allowing one person to talk into a machine and have another person's voice come out of the machine. This could have a wide variety of potential entertainment uses, and should be considered when evaluating the efficacy of speaker verification systems.

IV.E Non-Speech Applications

Although *Malcom's* power has been demonstrated by its ability to solve a very difficult speech analysis problem, the theory underlying *Malcom* is not restricted to speech. *Malcom* is ideally suited for many applications where something that is difficult to observe is moving smoothly and producing output that is a function of its position. For example, *Malcom* could be used to model the output sequence of sensors on a car engine or in a factory or in a nuclear power plant to determine whether the car/factory/power plant is operating normally. Various other applications involving tracking objects that are not easy to observe will undoubtably be found.

## V. First Year Deliverables

The goals for the first year are to create the tools necessary for studying *Malcom* and assess the potential of *Malcom* for speech recognition and low bit-rate speech coding.

V.A Tools

Two types of tools will be created for studying *Malcom*. The first of these will be a set of C (possibly C++) routines. These routines can be made available for use by other reseaerchers. In addition, since Entropic Research Laboratory's HTK speech analysis software is widely used by speech researchers, we will produce subroutines that can be used with the HMM toolkit to perform *Malcom*. The Entropic subroutines can also be distributed to other researchers.

V.B Speaker-dependent Diphone Recognition Results

While there are a wide variety of speech databases already available, the first attempt at speech recognition using *Malcom* will use data sets specifically designed to provide the transitions *Malcom* needs to infer the mapping between acoustics and articulation. Although the data sets required for *Malcom* are no more difficult to collect than current data sets, publicly available speaker-dependent isolated word databases (such as the Resource Management and TI 46-word isolated word databases) were not designed with *Malcom* in mind, and may not provide a sufficiently rich set of articulatory transitions. For comparison, the data set used in the preliminary experiment contained a rich set of vowel-to-vowel transitions and allowed the vowel space to be accurately mapped.

To ensure that sufficient articulatory transitions are available for each speaker, we will construct a data set for each speaker composed of at least two repetitions of $V_1CV_2$ utterances. The consonants will come from the set [p, b, t, d, k, g]. The vowel set will include 10 English vowels [i, ɪ, e, ɛ, æ, a, o, u, ʊ, ʌ]. This will provide 600 utterances (since $V_1$ can be the same as $V_2$), including 10 examples of each VC and and CV diphone.

Using the first repetition of each VCV, diphone models will be made using both *Malcom* and continuous density, left to right HMM's. Diphone recognition results for *Malcom* and for the HMM's will be obtained from the second repetition of each utterance. We expect the results to show that Malcom will outperform HMM's on this task.

### V.C Evaluation of Low Bit-Rate Speech Coding

As discussed in section V.A, in the process of creating word recognition models, *Malcom* calculates the number of bits that would be needed to transmit the speech if we converted the speech to articulator trajectories. Thus, we will also provide an estimate of the number of bits/second that would be saved by transmitting articulator positions instead of VQ codes.

## VI. Subsequent Year Deliverables

While we believe it is likely that *Malcom* will perform better than HMM's for speech recognition, it has not yet been demonstrtated that *Malcom* will provide sufficient accuracy for practical applications. To the extent that speech recognition work has been continuing for many years and is still inadequate, we see this research as high-risk work. However, we have also tried to make it clear that if Malcom does perform well, and we have reason to believe it will, the payoff could be very large. Because of high risk, we focussed on assessing the technology in the first year of research. Nonetheless, we have considered, and hope to study, modifications of *Malcom* that will have to be made to cope with more complex speech recognition tasks. Some possible extensions include: 1) using mixtures of Gaussians instead of simple Gaussians to represent the articulator distributions that produce the various sound types, 2) Allowing the diphone models to stretch or shrink to account for different speaking rates, 3) extending *Malcom* to multiple speakers by finding functions to convert the articulator-acoustic mapping for one person into the mapping for another (some techniques for doing so have already been devised but not tested).

In the area of speech coding, we hope to be able to use *Malcom* to encode voicing information as well as information about the vocal tract transfer function (which is essentially what we process now). This could make speech coding more efficient and further decrease the costs of sattelite communications.

## References

Ahalt, S., Krishnamurthy, A., Chen, P., & Melton, D. (1990). Competitive learning algorithms for vector quantization. Neural Networks, 3, 277-290.

Atal, B. S., Chang, J. J., Mathews, M. V., & Tukey, J. W. (1978). Inversion of articulatory-to-acoustic transformation in the vocal tract by a computer-sorting technique. Journal of the Acoustical Society of America, 63(5), 1535-1555.

Deng, L., & Sun, D. (1994). A statistical approach to automatic speech recognition using the atomic speech units constructed from overlapping articulatory features. Journal of the Acoustical Society of America, 95(5), 2702-2719.

Donovan, R. (1996). Trainable Speech Synthesis. Unpublished Ph.D. thesis, Cambridge University, Cambridge, U.K.

Erler, K., & Deng, L. (1992). HMM representation of quantized articulatory features for recognition of highly confusable words. ICASSP, 1, 545-548.

Fant, G. (1973). Chapter 1: The acoustics of speech, Speech Sounds and Features, . Cambridge, MA: MIT Press.

Gray, R. (1984). Vector Quantization. IEEE Acoustics, Speech, and Signal Processing Magazine, 4-29.

Hogden, J. (1995a). Computerized Voice Mimicry (White Paper ). Los Alamos, NM: Los Alamos National Laboratory.

Hogden, J. (1995b). Invention disclosure: A maximum likelihood approach to estimating articulator positions from speech acoustics . Los Alamos, NM: Los Alamos National Laboratory.

Hogden, J., Zlokarnik, I., Lofqvist, A., Gracco, V., Rubin, P., & Saltzman, E. (1996). Accurate recovery of articulator positions from acoustics -- new conclusions based on human data. Journal of the Acoustical Society of America, 100(3).

Huang, X. D., Ariki, Y., & Jack, M. (1990). Hidden Markov Models for Speech Recognition. Edinburgh: Edinburgh University Press.

Jordan, M., & Rumelhart, D. (1992). Forward models: supervised learning with a distal teacher. Cognitive Science, 16, 307-354.

Ladefoged, P., Harshman, R., Goldstein, L., & Rice, L. (1978). Generating vocal tract shapes from formant frequencies. Journal of the Acoustical Society of America, 64(4), 1027-1035.

Lee, K. F. (1989). Automatic Speech Recognition: The Development of the SPHINX System. Boston: Kluwer Academic Publishers.

Linde, Y., Buzo, A., & Gray, R. (1980). An algorithm for vector quantizer design. IEEE Transactions on Communications, COM-28, 84-95.

Markel, J., & Gray, A. (1976). Linear Prediction of Speech. New York: Springer-Verlag.

Marsden, J., & Tromba, A. (1981). Vector Calculus. (2 ed.). San Francisco: W. H. Freeman and Company.

McGowan, R., & Faber, A. (1996). Introduction to papers on speech recognition and perception from an articulatory view. Journal of the Acoustical Society of America, 99(3), 1680-1682.

McGowan, R., & Lee, M. (1996). Task dynamic and articulatory recovery of lip and velar approximations under model mismatch conditions. Journal of the Acoustical Society of America, 99(1), 595-608.

Mott, J. (1995). IPRB Preliminary Market Assessment . Los Alamos, NM: Los Alamos National Laboratory.

Muller, E., & McLeod, G. (1982). Perioral biomechanics and its relation to labial motor control. Journal of the Acoustical Society of America, 78(Suppl. 1), S38.

Nelson, W. (1977). Articulatory feature analysis -- I. Initial processing considerations. Memorandum, Bell Laboratories.

Neter, J., Wasserman, W., & Kutner, M. (1985). Applied Linear Statistical Models: Regression, Analysis of Variance, and Experimental Design. (second edition ed.). Homewood, Illinois: Richard D. Irwin, Inc.

Papcun, G., Hotchberg, J., Thomas, T., Laroche, F., Zacks, J., & Levy, S. (1992). Inferring articulation and recognizing gestures from acoustics with a neural network trained on X-ray microbeam data. Journal of the Acoustical Society of America, 92(2), 688-700.

Perkell, J., Cohen, M., Svirsky, M., Matthies, M., Garabieta, I., & Jackson, M. (1992). Electromagnetic midsagittal articulometer systems for transducing speech articulatory movements. Journal of the Acoustical Society of America, 92(6), 3078-3096.

Rabiner, L., & Juang, B. (1986). An introduction to hidden Markov models. IEEE ASSP Magazine.

Rose, R., Schroeter, J., & Sondhi, M. (1996). The potential role of speech production models in automatic speech recognition. Journal of the Acoustical Society of America, 99(3), 1699-1709.

Sayood, K. (1996). Introduction to Data Compression. San Francisco, CA: Morgan Kaufmann Publishers, Inc.

Schroeter, J., & Sondhi, M. (1992). Speech coding based on physiological models of speech production. In S. Furui & M. Sondhi (Eds.), Advances in Speech Signal Processing, (pp. 231-267). New York: Marcel Dekker, Inc.

Schroeter, J., & Sondhi, M. (1994). Techniques for estimating vocal-tract shapes from the speech signal. IEEE Transactions on Speech and Audio Processing, 2(1), 133-150.

Sondhi, M. (1979). Estimation of vocal tract areas: the need for acoustical measurements. IEEE trans. ASSP, 27(3), 268-273.

Wakita, H., & Gray, A. (1975). Numerical determination of the lip impedance and vocal tract area functions. IEEE trans. ASSP, 23(6), 574-580.

Zlokarnik, I. (1995). Adding articulatory features to acoustic features for automatic speech recognition. Journal of the Acoustical Society of America, 97(5 pt. 2), 3246(A).